Chapter Three: Contents

(Activity Generator – LA-UR-00-1725)

Chapter Three: Figures

Chapter Three: Tables

Chapter Three—Activity Generator

1. INTRODUCTION

1.1 Overview

The Activity Generator module generates a list of activities for each member of a synthetic population. Each activity consists of the following:

- activity type and its priority,
- starting and ending time preferences,
- a preferred mode of transportation,
- a vehicle preference (if appropriate),
- a list of possible locations for an activity, and
- a list of other participants (if the activity is shared).

The set of activities for each household is based on a household's demographics. They form the basis for determining individuals' trip plans for the region, thus resulting in travel demand sensitive to the demographics of a synthetic population. The activities also are sensitive to the network. Activity locations reflect land-use and employment data from the network. Feedback from the Route Planner or the Traffic Microsimulation provides network travel times.

1.2 Purpose

The Activity Generator has two principal purposes:

- To capture household behavior accurately—not just activity/travel patterns for individuals. Thus, if one family member takes a child to school, another need not do so.
- To ensure relative simplicity in the models for activity location. The location choice models are fairly simple. Instead of attempting to implement detailed models with network skim times etc., feedback from the Traffic Microsimulator and the Route Planner is used to refine activity location choices.

1.3 Activity Generator Major Input/Output

Fig. 1 shows how the Activity Generator uses synthetic population, survey, and network data to compute an activity list for every traveler.

The Activity Generator uses a synthetic population that has been located on a transportation network. The demographics in the population must match the demographics that are used in the Activity Generator. These demographics are used to select a suitable survey household activity pattern.

The activity survey is a representative sample of the population including travel and activity participation of all household members. Skeletal activity patterns are created by stripping locations from the survey.

The activity location table in a TRANSIMS network contains land use information that is necessary to determine the locations of activities derived from the skeletal survey patterns.

The Activity Generator's output consists of a data file that contains the activities for each traveler. This file identifies an individual and his/her corresponding household.

Each individual receives an activities list with the following attributes:

- type (e.g., home, work, shopping, school, or any other activity type collected in the survey),
- starting time range,
- ending time range,
- activity duration range,
- mode preference, and
- location.

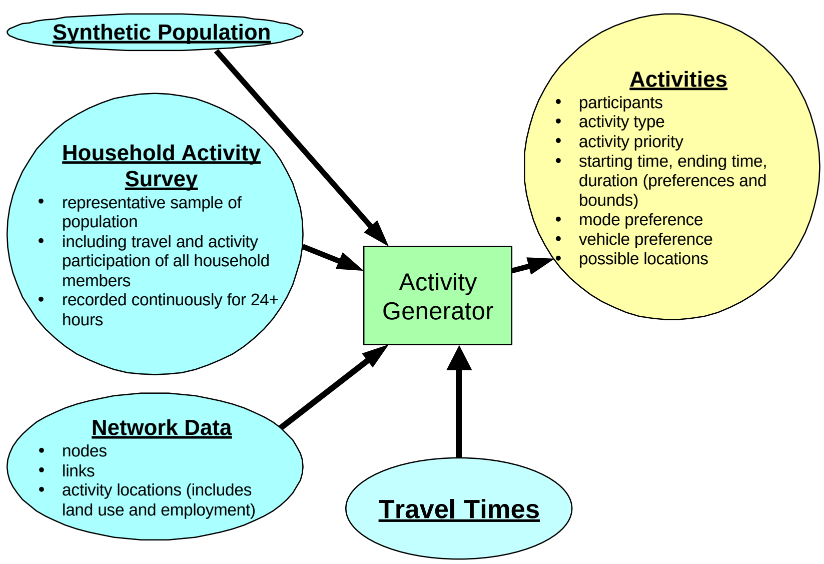

Fig. 1. The Activity Generator uses synthetic population, survey, and network data to compute an activity list for every traveler.

2. Module Description

2.1 Overview

TRANSIMS assumes that any two activities, separated by time and location, require travel between them. The degree of detail in both the synthetic population and activities can vary, depending on the availability of data and the study being performed. For example, a more detailed study with more realistic traffic requires a more detailed and realistic representation of the metropolitan population and the activities in which the population engages.

Each activity has a location; these locations must be a part of the TRANSIMS network data file. The location is one of the activity locations listed in the activity location network file. In addition to the time and location, a travel mode to reach the activity is assigned. If two or more individuals are making the same trip—in particular, a driving trip—the individuals are identified as part of the activity list.

2.2 Methodology

The Activity Generator overlays each synthetic household with a complete activity pattern. The following steps are used to create such a pattern.

Step One
- An activity survey is processed to obtain the total time spent in activity-by-activity type for each surveyed household.
- These times are weighted and summed to form a measure of total time spent in activities for each household.

Step Two
- Demographic variables of the household and the individuals in the household are selected based on which ones make the best predictors of the activity duration time.
- The predictor takes the form of a decision tree in which questions are asked at every level. The tree's terminal nodes are selected to be as homogenous as possible with respect to household activities.

Step Three
- Once a decision tree is constructed, each household from the survey is classified as belonging to one of the tree's terminal nodes.
- More than one household is usually assigned to each of the terminal nodes.

Step Four
- To allocate base activity patterns to individual households in the synthetic population, they are (1) classified according to the decision tree, and (2) given an activity pattern of one of the survey households

that were similarly classified as belonging to that node.

Step Five
- Drivers and passengers on shared trips within the household are identified. The skeletal activity pattern provides all necessary information for household interactions, including shared rides.

Step Six
- Initial travel times between activity locations are estimated by using average times or calculated using feedback from the Traffic Microsimulator or Route Planner for activity types and mode preferences.
- A modified discrete choice model based on land-use data (found in the activity location file) and travel times determines the locations of the activities, given the base activity pattern. Work locations are chosen first. Other activities are added using a multinominal logistic choice model.

2.3 Handling Freight and Itinerant Travelers

At this time, freight and itinerant travelers are handled in TRANSIMS through trip tables. The lack of data and models in these two areas makes this simplifying approximation necessary. This can be changed easily as better freight models are developed.

The following steps are taken to transform trip tables into TRANSIMS activities.

Step One
- Individuals (not created as part of synthetic populations) are created as vehicle drivers.

Step Two
- Locations for the start and end of the trips are chosen based on the zones specified in the trip tables.

Step Three One vehicle is created for each trip in the trip tables.

Note: There may be multiple trip tables representing various types of vehicles. For example, numerous trip tables could be used to represent trucks with different numbers of axles. A trip table must be available to accommodate the microsimulation of itinerant travelers.

In this version of TRANSIMS, a single vehicle is used for only one trip. If a new model generates a sequence of trips (e.g., for a single delivery truck), this would be implemented easily by creating an activity list with an activity at every stop.

The vehicle for each trip is listed in the TRANSIMS vehicle file. Each vehicle is assigned an emissions type.

3. Algorithm

3.1 Approach

The activity generation algorithm consists of four principal steps. Fig. 2 shows these steps graphically.

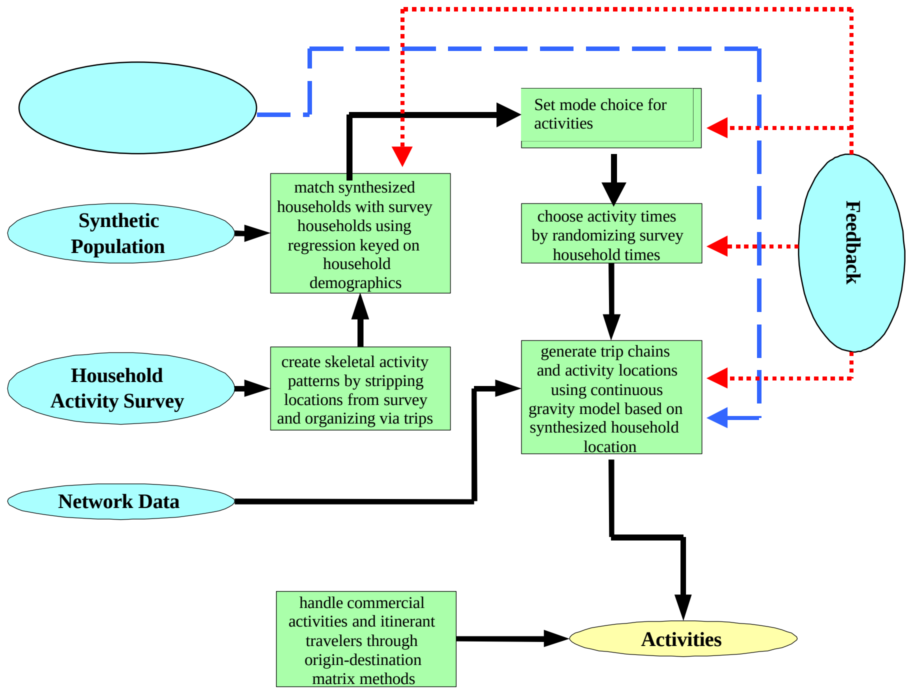

Fig. 2. The activity generation algorithm.

Step One

- Create skeletal patterns from the survey.
- Organize by trips the activity list for each survey household.
- Strip locations.
- The final product consists of a library of skeletal activity/travel patterns.

Step Two

- Match synthesized households with survey households.
- To make matches, use a tree-structured classification based on household demographics.
- Assign each synthetic household to a unique node in the tree.

- After the synthesized household is assigned to its tree node, select a survey household at random relative to the weights given to the survey households from the same node to obtain a matching household.

- Assign the skeletal patterns for the survey household members to the matching members in the synthesized households.

Step Three

- Group household members to trips. The final activity set includes a list of participants for each activity.

Step Four

- Because households are matched on demographics, an activity list for a person in the matching survey household is appropriate for a person in the reconstructed household except for activity location.

- Use the synthesized household location and choice models to generate new locations for the activities.

The synthetic population household demographics, survey household demographics, survey household activity/travel patterns, and a binary matching tree are inputs to the first stage, producing a skeletal activity/travel pattern for the synthesized household. Household members are matched to trips, and network land-use data are then used to generate activity locations for the skeletal pattern and create the final activity list.

3.2 Binary Tree Matching

Cross-tabulation of households by demographic variables can easily create many cells with few or no households in the survey. Instead of matching through some kind of table, household matching locates households in the terminal nodes of a binary tree. Although this tree is sensitive to the characteristics of household behavior, it is also parsimonious with respect to household characteristics that do not affect behavior.

3.3 Sample Binary Matching Tree

Fig. 3 shows a simple sample of a binary matching tree. Actual trees used in generating activities are much more complicated and will be specific to each particular transportation study.

The tree shown in Fig. 3 was constructed with an abbreviated list of three household demographic variables:

hhsize	household size
agelt5	number in household aged less than 5

age5to17 | number in household aged from 5 to 17

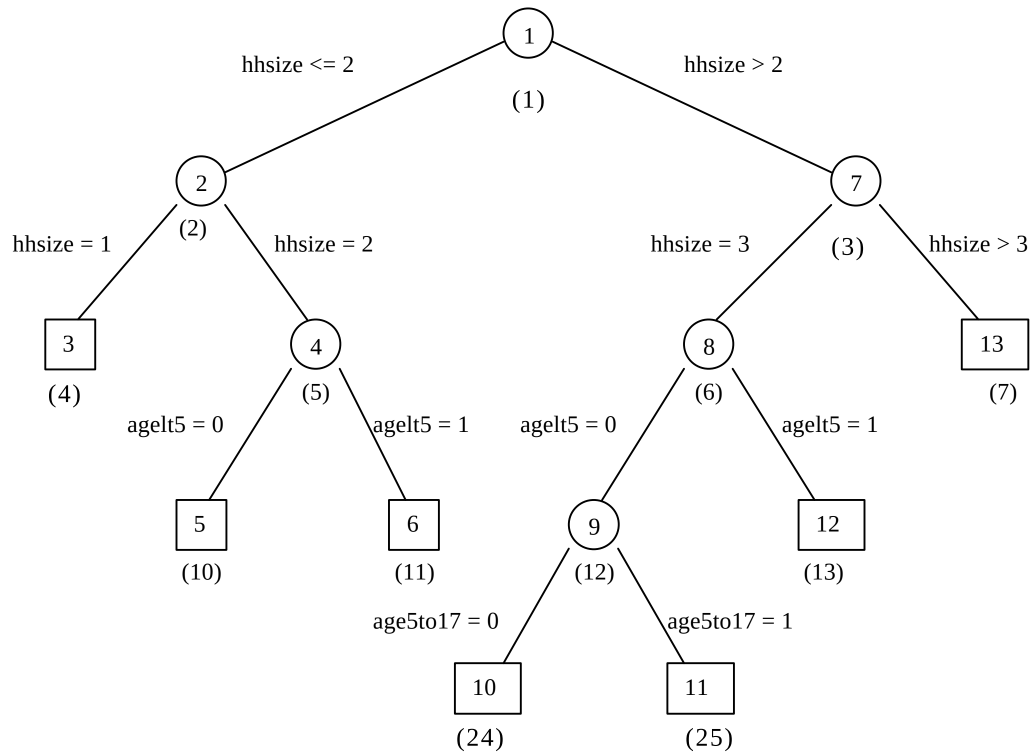

Fig. 3. This tree has 13 nodes, six of which are nonterminal nodes indicated by circles and seven are terminal nodes indicated by squares. The numbers inside the squares and circles are node numbers. The numbers beneath the nodes in parentheses are binary node numbers.

The tree in Fig. 3 has 13 nodes, including seven terminal nodes indicated by squares. Table 1 defines these seven terminal nodes.

At each non-terminal node, a household is classified further into either a left or right "child" node according to a simple rule given by a demographic variable and split point. If the appropriate variable is less than the split point, the household is classified into the left node; otherwise the right node is selected.

In the example above, the first choice (node 1) is on household size (*hhsize*). If *hhsize* is less than 2.5, the household falls somewhere in the left nodes. Otherwise, further classification proceeds to the right. The procedure continues recursively until a terminal node is reached.

Table 1. Terminal nodes.

Node	Description
3	Household size = 1
5	Household size = 2, no children less than 5
6	Household size = 2, 1 child less than 5
10	Household size = 3, no children less than 5, no children between 5 and 17
11	Household size = 3, no children less than 5, 1 child between 5 and 17
12	Household size = 3, 1 child less than 5
13	Household size greater than 3

3.4 Generating an Activities Set

The first step in generating a set of activities is to locate the synthetic household in its unique terminal node of the tree. A survey household is then selected at random from the node. For flexibility, weights are used in choosing the survey household. Each survey household has weight w_i assigned to it in the Survey Weights file. If N is the terminal node for the synthetic household, survey household i in node N is chosen with the following probability:

$$p(i) = \frac{w_i}{\sum_{j \in N} w_j}.$$

3.5 Matching Individuals Within the Household

Synthetic household members are matched to members in the survey household based on the four demographic variables.

Relate	Variable coded for relation to head of household 1 = spouse or partner, head of household PUMS variable RELATE1 = 00, 01, 10 2 = child RELATE1 = 02, 03, 06 3 = adult relative RELATE1 = 04, 05, 07 4 = other
Work	Work status 1 = full-time/part-time worker (including self employed) 2 = nonworker
Gender	1 = male 2 = female
Age	Age in years

These four variables must be in the person demographics in both the synthetic population file and the survey demographic file.

3.6 Making the Best Possible Matches

Although perfect matches are not always possible, the Activity Generator does attempt to find the best matches possible. Moreover, it sets a flag to resample from the pool of survey households in the matching node if certain conditions hold.

3.6.1 Rules for Children

- Children are matched to children, and adults are matched to adults.
- The children in the synthetic household are sorted by gender and age (descending). Children in the survey household are sorted in the same way, and a one-to-one match is made between the two sorted lists.
- If the survey household has more children than the synthetic household, the extra children in the survey household are ignored.
- If the synthetic household calls for more children than the survey household has, the activities of the last child in the survey household are replicated as often as necessary to create activities for the remaining children in the synthetic household.
- If there are no children in the survey household but the synthetic household has children, the error flag is set to resample from the survey households and try a different match.

3.6.2 Rules for Adults

- The adults in the synthetic household and the survey household are sorted by the following variables: relate, work, gender, and age (descending sort).
- Again, a one-to-one match is made.
- If the synthetic household contains more adults than the survey household, the activities of the last adult survey household member are replicated as often as necessary.
- At present, no further error checking is performed in the matching algorithm for adults.

3.7 Multivariate Regression Tree Program

A multivariate regression tree program can help construct a binary tree that matches synthetic households to survey households. This section provides a brief description; for a more detailed description, consult the paper by Vaughn, Speckman, and Sun[1].

A regression tree is a technique for modeling a regression relationship between a dependent variable Y and independent variables $X_1, X_2,\ldots,X_p$. Regression trees are useful when there are a large number of explanatory variables and there is an expected complex relationship between the response variable and the explanatory variables.

In these cases, tree-based methods may more easily capture non-additive behavior and thus be easier to interpret than linear models. The CART (Classification and Regression Trees) algorithm was introduced by Breiman et al.[2] and has been implemented as the tree function in the S-PLUS software package (see the paper by Clark and Pregibon[3]). The basic idea is to partition the data set into nodes defined by the predictor variables X_1, $X_2,\ldots,X_p$, and to model the response as a constant within each node. A binary tree defines these nodes.

As implemented by the CART algorithm, tree modeling consists of two stages:

- a forward recursive algorithm for "growing" the tree, and
- a second stage where the tree is "pruned back."

Because the growing process is only in the forward direction (once a node is defined, it cannot change), the algorithm does not necessarily produce an optimal tree. The strategy is to begin by growing a very large tree—one that probably "overfits" the data—then to use a second algorithm, thus balancing faithfulness to the data with the complexity of the tree to select a good subtree.

The philosophy is related to forward selection in the usual regression setup, where a liberal rule is adopted in the entering of variables to ensure that no important variables are omitted.

3.8 CART Algorithm

The following two sections describe how the CART algorithm grows and prunes trees.

3.8.1 Growing the Tree

To define the recursive algorithm, consider observations in a single "parent" node P_I as part of tree T. At the next stage, the parent node is split into two "children" nodes:

- a left node, $L_{I'}$, defined as all I' observations in P_I with $X_{ij} \le t$, and

[1] Vaughn, K.M., Speckman, P. and Sun, D. (1999) *Identifying Relevant Socio-Demographics for Distinguishing Household Activity-Travel Patterns: A Multivariate CART Approach*.
[2] Breiman, L., J.H. Friedman, R.A. Olshen and C.J. Stone. (1984) *Classification and Regression Trees*. Chapman and Hall, London.
[3] Clark, L.A., and Pregibon, S. (1990) *Tree-based models.* In Statistical models in S. Wadsworth and Brooks/Cole, pp 377-419.

- a right node, R_I”, where $X_{ij} > t$ for a suitable choice of variable X_{ij} and cut point t.

The optimal variable and cut point for the split are defined in terms of the “deviance” of the node, given as

$$D(N) = \sum_{i \in N} (Y_i - \overline{Y})^2 ,$$

the corrected total sum of squares for the observations in the node. For a potential partition split on variable X_j at cut point t, define the reduction in deviance from the split as

$$\Delta_{j,t} = D(P) - (D(L) + D(R)).$$

A search is conducted over all j and t to find the pair j^*, t^* such that

$$\Delta_{j^*,t^*} = \max_{j,t} (\Delta_{j,t}) ,$$

subject to,

- I', I'' > some minimum (say 10)
- D(P) > 0.01*D(total),

where *D(total)* is the deviance in the dependent variable before any splits are made. If either condition fails, the parent node is a terminal node. The algorithm recursively split nodes until they all become terminal nodes.

3.8.2 Pruning the Tree

Prediction for a regression tree begins when the dependent variable Y is estimated by the mean value of Y in each node. However, the binary tree from the growing algorithm generally overfits the data.

Several proposals have been made to determine a better tree. One common way to assess how well a tree fits is by using it to predict a new set of data. In this case, deviance is replaced by a sum-squared prediction error. It is from this that the best subtree in the sense of minimizing prediction error can be determined. The selected tree partially depends on the data set selected to be held out. Holding out such a subset for validation may prove wasteful.

Following Breiman et al. (1984), S-PLUS implements a form of cross-validation that emulates this kind of validation without wasting data. The data set is randomly partitioned into ten approximately equal parts. Each part is held out in turn. A subtree T’ is then re-estimated on the remaining 90% of the data, and the re-estimated tree is used to forecast the 10% held out.

Let $CV_i(T')$ denote the sum squared prediction error for the ith partition. The process is repeated for all ten subsets of the data, and a total cross-validation score,

$$CV(T') = \sum_i CV_i(T'),$$

is computed for the subtree. A subtree that minimizes (or nearly minimizes) $CV(T')$ is a good final choice for a tree that is appropriate for the data.

3.9 Multivariate Tree Algorithm

The following extended definition of deviance is used to implement the multivariate regression tree: Suppose we have dependent variables $Y_1,\ldots,Y_d$ and node N with I' observations.

Let the deviance at node N with respect variable Y_j be given as

$$D_j(N) = \sum_{i'} (Y_{i'j} - \bar{Y}_j)^2,$$

with the total deviance at node N

$$D(N) = \sum_j s_j D_j(N) = \sum_j \sum_{i'} s_j (Y_{i'j} - \bar{Y}_j)^2,$$

where $s_j = 1/\mathrm{var}(Y_j)$ and is a scale factor. Then for a tree T,

$$D(T) = \sum_j s_j D_j(T) = \sum_j \sum_i s_j (Y_{ij} - \bar{Y}_j)^2$$

is the scaled total sum of squares for the I observations in the tree. Nodes now can be split by using total deviance instead of the single-variable deviance. With this new definition of total deviance, a regression tree can be grown and pruned as before. When coupled with cross-validation, this definition of deviance can be used to prune a tree to a proper size.

In the Activity Generator, the trees are constructed using the total times households spend in 15 broadly classified activity types as the values of Y_{ij}. An additional Y value is the number of trips the household makes. The predictor variables are all of the demographic variables collected in the survey and the housing density. While these are reasonable variables to construct the tree, any variables could be used.

3.10 Activity Location Generation

The following steps show how the location generation algorithm works. Locations are selected by following two steps. First, a discrete choice model is used to select appropriate zones for all activities. And second, land-use variables are used to find specific activity locations within zones for each activity.

Step One Use a discrete choice model to generate all work locations.

Step Two Use trip-chaining discrete choice models to generate locations for other activities.

3.10.1 Work Location Model

The work location model is a simplified multinomial logistic choice model, defined with the following terms.

L Destination zone for work activity.

a(L) Attractor for work activity in zone *L*. Often expressed as *a(L) = c'x(L)*, where *x(L)* is a vector of land-use variables for zone *L*, and *c* is a coefficient vector fit by maximum likelihood. It is also possible to use other specifications for *a(L)*, including a nonparametric model for a continuous distribution fit from data. This model is described in a technical report by Speckman, Sun, and Vaughn [1998].

t(H,L) Travel time from home location *H* to work location *L*.

b_m Coefficient for mode choice *m*.

The destination zone is selected according to the probability distribution

$$p(L) = \frac{\exp(a(L) + b_m t(H,L))}{\sum_{L'} \exp(a(L') + b_m t(H,L'))}.$$

Initial mode choice is taken from the survey household skeletal pattern. After the zone is selected, a specific activity location within the zone is selected at random according to weights given in the Activity Location file. For work, employment weight w_A for each location is used, and location A is selected with probability

$$p(A) = \frac{w_A}{\sum w_{A'}},$$

where the sum is taken over all locations *A'* in zone *L*.

This is a simple model for several reasons. Because TRANSIMS starts with an empty network, the Activity Generator may not have access to realistic travel times. Network travel times can be fed back from the Route Planner or the Traffic Microsimulator to the Activity Generator to refine the location choice probability model. With these travel time updates, the Selector/Iteration Database module is used to develop models for mode and location choice.

3.10.2 Locations for Other Activities

To generate locations for other activities, we use a logistic multinomial choice chaining model. For example, consider a trip chain that consists of two stops on the way from work to home.

Suppose the skeletal pattern calls for travel from work location *L* to "visit" at L_1 by mode m_1, a second stop to "shop" at location L_2 by mode m_2, and finally returning home by mode m_3. The home and work locations are known. The other two destination zones, L_1 and L_2, are determined successively. For the first location, L_1, the work location *L* and the home location *H* are used in the following equation as the anchor locations of the trip

$$p(L_1) = \frac{\exp(b_{m1}d(L,L_1) + a(L_1,v) + b_{m2}d(L_1,H))}{\sum \exp(b_{m1}d(L,L_1') + a(L_1',v) + b_{m2}d(L_1',H))},$$

where the sum is over all zones. After L_1 is chosen, it will replace the work location *L* as the first anchor location for choosing the shop location L_2. In this example, separate attractors *a(L, t)* are defined for each location *L* and activity type *t*, where *t* is either *v* for "visit" or *s* for "shop."

<u>Example:</u>

As an example, we generated an activities set for a two-adult household in Portland, Oregon. The node in the matching tree consisted of all households with two adult workers. There were 656 survey households in the node, and one was selected at random as a match. The first person had a single work activity, followed by

- a stop to visit,
- a stop to shop,
- a maintenance stop (car wash), and
- a return home.

Fig. 4 shows the set of generated activities on a map with the EMME/2 Portland network.

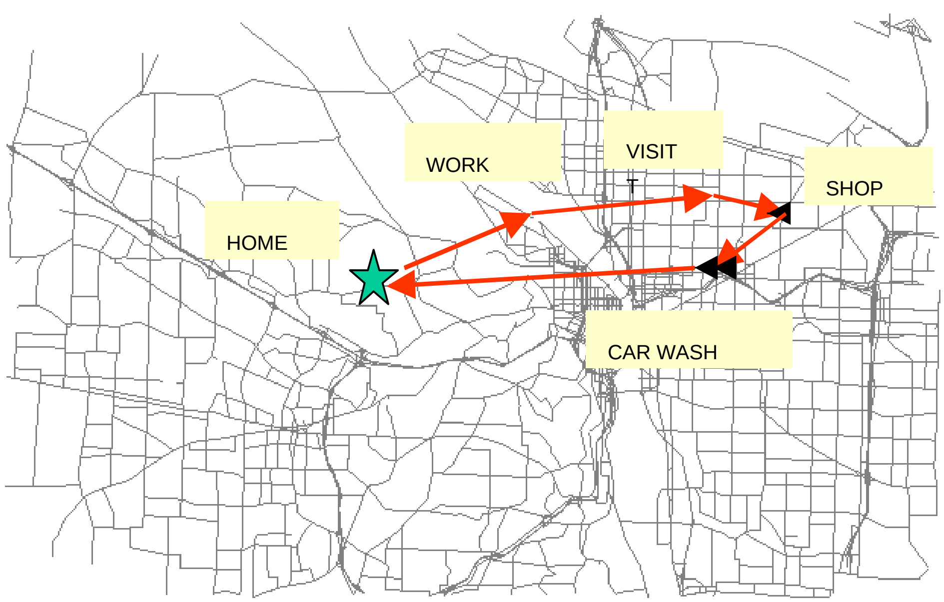

Fig. 4. Activities for the first traveler.

A second person has the following schedule: The top began from home to work, with a stop on the way, then went out of town on a business trip (coded "other work" in the survey), returned to the office, and finally went home.

The activity list currently has no code for out-of-town business, so as default the Activity Generator selected a second work location for the "other work" activity. The survey also had incomplete information on the last work activity with a missing location. The Activity Generator created a third location for this activity. The resulting activity pattern is shown in Fig. 5.

This example demonstrated that the Activity Generator functions well even with missing data in the survey record.

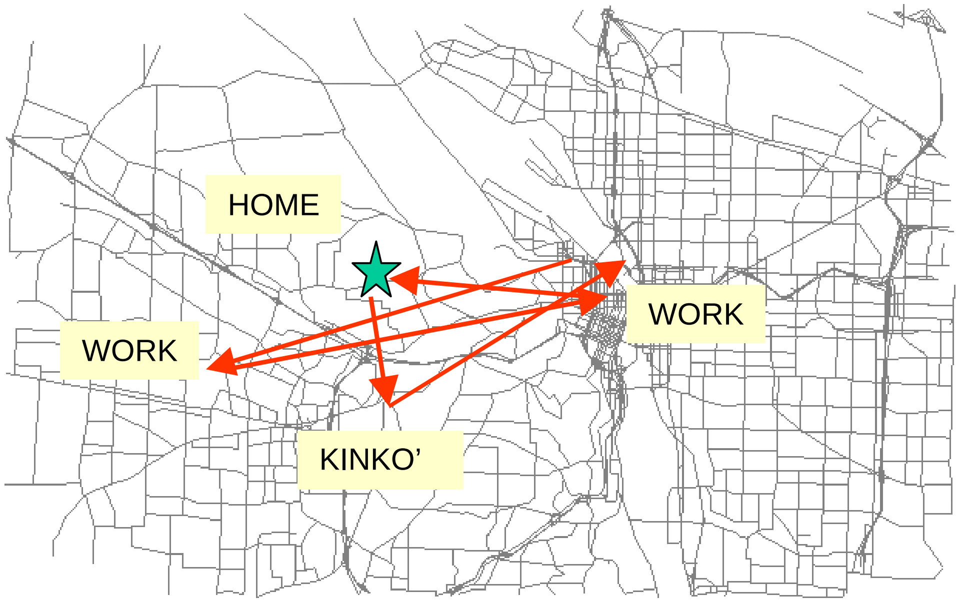

Fig. 5. Activities for the second traveler.

3.11 Activity Times and Durations

Activity times are taken from skeletal activity patterns with as little change as possible for out-of-home activities. Each activity has a preferred start time, end time, and duration. The range of each of these times is specified by a beta distribution, which requires four parameters:

- lower bound L,
- upper bound U, and
- "shape" parameters a and b.

When $a = 1$ and $b = 1$, no preference is indicated within the range L to U. If $a = -1$ and $b = -1$, the range is assumed to be 0 around the preferred time. Preferred times are taken directly from the skeletal patterns. Table 2 gives the values of the remaining parameters as currently implemented in the Activity Generator. The actual travel times between two activities given by this method may be infeasible. Using output from the Traffic Microsimulator or the Route Planner, the Selector/Iteration Database can be made to request new times for these activities.

Table 2. Settings of time parameters for activities. Obs. means observed time taken from skeletal pattern. The times are in hours.

Type of activity		L	U	a	b
Work	Start	Obs. - .25	Obs. + .25	1	1
	End	Obs. - .25	Obs. + .25	1	1
	Duration	Obs. - .25	Obs. + .25	1	1
Other out-of-home	Start	Obs. - .50	Obs. + .50	1	1
	End	Obs. + .50	Obs. + .50	1	1
	Duration	Obs. - .3(obs.)	Obs + .3(obs.)	1	1
AM beginning at-home	Start	0	0	-1	-1
	End	Obs. - .75	Obs. + .75	1	1
	Duration	Obs. - .75	Obs. + .75	1	1
Home during day	Start	Obs. - .75	Obs. + .75	1	1
	End	Obs. - .75	Obs. + .75	1	1
	Duration	Obs. – 1.00	Obs. + 1.00	1	1
PM end at-home	Start	Obs. - .75	Obs. + .75	1	1
	End	24.00	24.00	-1	-1
	Duration	Obs. - .75	Obs. + .75	1	1

4. Activity Regeneration

4.1 Feedback to the Household Activity List

TRANSIMS is designed so that the Selector/Iteration Database module using feedback from the Route Planner and Traffic Microsimulator modules can easily request changes in activity mode preferences, times, and locations. There is a tradeoff between spending resources to make the original activity list as correct as possible and correcting the list by feedback. This is an active area of TRANSIMS research. Using feedback, it is possible to begin with a "rough" base activity list. Once the feedback begins, a refined list will emerge.

The importance of feedback to refine the activity list should not be underestimated. If feedback can improve an activity list so that extremely detailed activity models are not necessary, then national survey data and data from other sources could be used to construct the base set of activities. In this case, an extensive local survey would not be needed.

When the Route Planner or Traffic Microsimulator detects a set of impossible activities, new locations, mode preferences, and activity times can be generated or the entire household's set of activities can be regenerated. The Activity Regenerator program is used to change the existing activity list by partial regeneration of the activities or to generate a new activity list for the entire household. Feedback can also provide an updated list of travel times to be used in the activity regeneration. A file of feedback commands is used by the Activity Regenerator to specify the activity to be updated and the type of update to be applied.

4.2 Feedback Commands

Regenerate the household's activity list. Regeneration is accomplished by rematching the synthetic household to a survey household in the regression tree to select another activity pattern for the household.

- Change the location for the activity.
- Change the mode for an activity.
- Change the mode to the specified mode and choose a new location for the activity.
- Change the time for the activity. A new start time, end time, and alpha and beta parameter for the time distribution can be specified.

5. FILES

5.1 Input Files

5.1.1 Binary Tree

A file specifies the tree structure. The variables in the tree structure are chosen as described in Section 3.3. The demographic variables used to define the tree are assumed to be the variables `<HHdata1 … HHdatan>` in the synthetic population file. The order of appearance in the synthetic population file determines the numbering of the variables in the tree—this order must match the variables in the survey population file.

These variables may come directly from census classifications or may be constructed from the survey and data records of synthetic population household members. Variables are numbered 1-n with 0 used to indicate a leaf node in the tree.

File Format:

The file format has one line for each node with the following format:

```
<variable> <splitvalue> <binary node number>
```

Example:

A simple tree that assumes the following demographics for both the synthetic population and the survey population files:

```
Household demographics: hhsize alt5 a5to17
```

Variable 1 is `hhsize`, variable 2 is `alt5` (age < 5), and variable 3 is `a5to17` (ages 5 to 17). The following file gives the complete specification of the tree shown in Figure 3:

```
1  2.5    1
1  1.5    2
0  0      4
1  0.5    5
0  0     10
0  0     11
1  3.5    3
2  0.5    6
3  0.5   12
0  0     24
0  0     25
0  0     13
0  0      7
```

5.1.2 Survey Weights

The file containing the weights has the relative weights for the survey households in the regression tree.

Format of the file is:

```
<survey household id> <weight>
```

The weights are floating point numbers.

5.1.3 Survey Household Demographic Data

The Activity Generator uses a set of demographic data for the survey population corresponding closely to the data for the synthetic population. The survey demographic data is contained in a TRANSIMS synthetic population file format. The tract and block group entries are set to zero, and the location is set to –1.

The household demographic/data variables must exactly match the variables used for the tree-matching algorithm. The person demographics must have the variables `RELATE`, `WORK`, `GENDER` and `AGE`.

Example:

`Household Demographics: HHSIZE ALT5 A5TO17`

Person Demographics: `RELATE WORK GENDER AGE`

The household data lines in a file with this header will have three household demographic values: (`HHSIZE`, `ALT5` and `A5TO17`). The person demographic data lines will have the four required variables: `RELATE`, `WORK`, `GENDER` and `AGE`.

5.1.4 Survey Activities

To generate skeletal household activity patterns, we use the set of activities from the survey households. The activity list must match the survey household demographics file. Each household and person in the demographics file must appear in the same order with activities in the activity file.

File Format:

Activities are grouped sequentially for each survey household. The file consists of an ASCII file with entries separated by one or more spaces. The first line defines the column headings for the file.

Example:

The following set of activities is for household # 200090, a household with two persons. The first person has 13 activities, including an initial at-home activity. Activities 2, 3, and 7 are out-of-home activities. The second person has two out-of-home activities.

SAMPNO	PERSNO	ACTNO	ACTID	AT_HOME	WUTHERE	MODE	DRIVER	NUNVEH	ACTSTART	ACTEND	GEOX	GEOY
200090	1	0	0	1	2	1	0	0	0	480	7651244.0000	668103.5625
200090	1	1	0	1	2	2	0	1	480	540	7651244.0000	668103.5625
200090	1	2	2	2	2	2	1	2	550	645	7655157.5000	644216.5625
200090	1	3	2	2	2	2	1	2	650	690	7654753.0000	645012.8750
200090	1	4	5	1	2	2	1	2	700	720	7651244.0000	668103.5625
200090	1	5	5	1	2	2	0	1	720	765	7651244.0000	668103.5625
200090	1	6	0	1	2	2	0	1	780	900	7651244.0000	668103.5625
200090	1	7	2	2	2	2	1	1	910	960	7655037.0000	644165.3125
200090	1	8	0	1	2	2	1	1	970	1050	7651244.0000	668103.5625

200090	1	9	4	1	2	2	0	1	1050	1095	7651244.0000	668103.5625
200090	1	10	5	1	2	2	0	1	1095	1140	7651244.0000	668103.5625
200090	1	11	0	1	2	2	0	1	1140	1303	7651244.0000	668103.5625
200090	1	12	0	1	2	2	0	1	1303	1620	7651244.0000	668103.5625
200090	2	0	0	1	2	1	0	0	0	480	7651244.0000	668103.5625
200090	2	1	0	1	2	2	0	1	480	540	7651244.0000	668103.5625
200090	2	2	2	2	2	2	2	2	550	645	7655157.5000	644216.5625
200090	2	3	2	2	2	2	2	2	650	690	7654753.0000	645012.8750
200090	2	4	5	1	2	2	2	2	700	720	7651244.0000	668103.5625
200090	2	5	5	1	2	2	0	1	720	765	7651244.0000	668103.5625
200090	2	6	0	1	2	2	0	1	780	900	7651244.0000	668103.5625
200090	2	7	5	1	2	2	0	1	900	1080	7651244.0000	668103.5625
200090	2	8	5	1	2	2	0	1	1080	1170	7651244.0000	668103.5625
200090	2	9	5	1	2	2	0	1	1170	1200	7651244.0000	668103.5625
200090	2	10	0	1	2	2	0	1	1200	1310	7651244.0000	668103.5625
200090	2	11	0	1	2	2	0	1	1310	1620	76512440000	668103.5625

Table 3. Survey Activities file format.

Field	Description	Allowed Values
Survey Household ID	Each survey household has a unique ID.	integer
Person Number	Each person in the household has a unique ID, starting with 1. Numbers are only unique within the household.	integer: 1 through household size
Activity Number	Activity number for each person.	integer: 0 through n: 0 = initial at-home activity of the day if necessary
Activity Type	Definitions may vary.	integer: 0 through n: Example: 0 = at-home activity 1 = work 2 = shop 3 = school 4 = visit 5 = other 6 = serve passenger
At Home	Coded 1 for activity at-home, 2 for out of home.	integer: 1 or 2
Were-you-there	Coded 1 if person was already at the location, 2 if not.	integer: 1 or 2
Mode for arriving at activity	Integer code for mode. Modes must correspond to modes in TRANSIMS mode map file used by the Route Planner and the Traffic Microsimulator.	integer: 1 through n: 1 = Walk 2 = Car 3 = Transit 4 = Light Rail 5 = Park and Ride outgoing 6 = Park and Ride incoming 7 = Bicycle 8 = Magic Move
Driver	Coded 1 if person was the driver, 2 if passenger. Otherwise, 0.	integer: 1, 2, or 0
Activity Start Time	Time of start of activity in minutes after midnight.	integer: 0 through 2400
Activity End Time	Time of end of activity in minutes after midnight.	integer: 0 through 2400
Number in Vehicle	Number of people in vehicle	integer:1 through n
Geocode x	Easting geocoordinate. Must be in units agreeing with mode coefficients.	decimal

Field	Description	Allowed Values
Geocode y	Northing geocoordinate. Must be in units agreeing with mode coefficients.	decimal

5.1.5 Zone Data

The zone data contain geographic data for zones used in the Activity Generator and the attractors by zone and activity type used in the location choice models. These data may be aggregated from the land-use data associated with the activity locations within the zone. These could include, for example, the total employment by SIC code or the total square footage of retail property in the zone. It should be noted that the zones could be as large as usual Traffic Analysis Zones or as small as the individual activity locations. The computation time however increases significantly as the number of zones increases.

Format:

There is one header line with definitions of the variables. Each variable has a one-word descriptor with no white space. The format is as follows:

```
zone NORTHING EASTING <attractor1> … <attractorN>
```

The attractor variables must correspond to the activity types in the Activity Generator. The attractors are for activity types 1 – N.

Example:

The file below gives zone data for a test network with 10 zones. Six activity types are defined: `work, shop, school, visit, other,` and `serve_pass.`

ZONE	EASTING	NORTHING	WORK	SHOP	SCHOOL	VISIT	OTHER	SERVE_PASS
1	50	16050	2	2	-10	-10	2	1
2	50	19050	2	2	-10	-10	2	1
3	16050	35050	2	2	-10	-10	2	1
4	19050	35050	2	2	-10	-10	2	1
5	35050	19050	2	2	-10	-10	2	1
6	35050	16050	2	2	-10	-10	2	1
7	19050	50	2	2	-10	-10	2	1
8	16050	50	2	2	-10	-10	2	1
9	17500	17500	-10	-10	-10	-10	-10	1
10	17500	17500	-10	-10	3.71	3.71	–10	1

5.1.6 Mode Coefficient Data

The coefficients b_m for distance in the choice models depend on mode *m*. This file contains coefficients by activity type by mode. These coefficients are determined by a logit fit to a simple discrete choice model given by

$$p(L) = \frac{\exp(a(L) + b_m t(H,L))}{\sum_{L'} \exp(a(L') + b_m t(H,L'))}.$$

<u>Format</u>:

The format of this file is

```
<coefficient> <activity type> <mode number>
```

Every activity type and mode used in the Activity Generator must have an entry in this file including the home activity type.

<u>Example</u> for activity types 0 to 6 and modes 1 and 3:

```
-0.000080     1     1
-0.000081     1
-0.000082     1
-0.000083     1
-0.000084     1
-0.000085     1
-0.000086     1
-0.000040     0     2
-0.000040     1     2
-0.000040     2     2
-0.000040     3     2
-0.000040     4     2
-0.000040     5     2
-0.000040     6     2
```

5.1.7 Feedback Commands

The feedback command file contains commands that tell the Activity Regenerator which activities to regenerate and what action to take for each specified activity. The Household ID and Activity ID must always be specified. The command is a string that may be followed by optional command parameters. Each command is on a separate line in the file.

Format of this file is:

```
<Household Id> <Activity Id> <Command> [<command parameters>]
```

Commands that are currently supported are:

- `L` – change the location for the activity.
- `M <mode value>` – change the mode for the activity to the integer mode value.
- `LM <mode value>` – change the mode for the activity to the mode value and then change the location for the activity.

- `T <start time> [<end time>] [<alpha parameter>] [<beta parameter>]` – change the time for the activity to start time and if specified, end time with alpha and beta parameters on the time range.
- `R` – regenerate the entire activity list for the household by rematching with a survey household.

Example:
```
1356   7  L
1358   2  LM  3
1379  10  M   2
1386   4  T   420  1040  0.5  1.0
1395   R
```

In the example above,

- Activity 7 in Household 1356 will have a new location generated.
- Activity 2 in Household 1358 will have the mode changed to 3 and then a new location generated.
- Activity 10 in Household 1379 will have the mode changed to 2.
- Activity 4 in Household 1386 will have the start time changed to 420, the end time changed to 1040, the alpha parameter set to 0.5, and the beta parameter set to 1.0.
- Household 1395 will have all of the activities regenerated.

5.1.8 Travel Times

The travel times file contains information about the travel times between zones and provides a mechanism to update the travel times used in the Activity Generator. If the travel time between a zone pair is not present in the travel times file or if no travel time file is specified, the Activity Generator uses a default travel time for the zone pair. The travel times in the file are specified by mode and a time of day range.

Format of this file is:

```
<zone1> <zone2> <mode number> <start time> <end time> <travel time> <last update>
```

- zone 1 – the number of the first zone.
- zone 2 – the number of the second zone.
- mode number – the TRANSIMS mode number. This number must be the same as the modes used in the Activity Generator and Route Planner.
- start time – the first time in the time of day range in minutes since midnight.
- end time – the last time in the time of day range in minutes since midnight.

- travel time – the travel time for the zone pair in seconds
- last update – time this entry was updated in seconds since midnight

Example:

```
3   7   2   300   600   900   18000
```

The travel time between zones 3 and 7 for mode 2 between the hours of 5 a.m. and 10 a.m. is 15 minutes (900 seconds). This entry was last updated at simulation time of 5 a.m. (18000 seconds).

5.2 Output Files

5.2.1 Activity Files

The following activity file is the protocol for the interaction of the TRANSIMS activity sets with the TRANSIMS Route Planner and Traffic Microsimulator.

File Format:

The activity file consists of an ASCII file containing the activity data. Activities for a household are grouped sequentially in the activity file. Each line of the file contains tab-delimited data fields for a single activity.

Appendix A provides the meaning and format of the activity data fields. For most fields, the entry *–1* denotes an unspecified value.

The reference time is taken as 0.00 (midnight of the first day). All times are decimal numbers that denote the number of hours from 0.00. Note that each time should be given to a minimum of

- two decimal places to capture minutes, and
- four decimal places if seconds are necessary.

Each activity has a start time, end time, and duration range. The preferred time for each of these is given in terms of the two parameters of a beta distribution,

$$f(t) = C(t-L)^{a-1}(U-t)^{b-1},$$

where

- C is a constant,
- L is the lower bound of the time,
- U is the upper bound, and
- a and b are the parameters that specify the distribution.

The mean of the distribution is $\frac{a}{a+b}$; $a=1$ and $b=1$ gives a uniform distribution between L and U, and the larger a and b are, the more peaked the distribution.

5.2.2 Problem File

The Activity Generator writes entries to a problem file when certain conditions are encountered during activity generation for a household. These conditions include

- insufficient vehicles in the household for the driving trips,
- failure to find a suitable driver for intra-household shared rides, and
- incomplete match between the survey household and the synthetic household.

The name of the problem file is specified with a configuration file key—`ACT_PROBLEM_FILE`.

File Format:

The problem file is composed of entries, one per line, that contain an integer denoting a problem type, the number of optional fields to follow, and information specific to the type of problem. The problem types are:

1) failed to find driver for shared ride
2) incomplete match between survey household and synthetic household
3) arrived too late at this activity
4) failure to adjust times for shared ride activity
5) unable to assign a vehicle for a driving trip

The problem types are defined as an enumeration (`ActivityProblemType`) in *ACT/ActivityProblem.h*.

Format for types 1 and 5 is
```
   <type> 3<HH Id> <Person Id> <Activity Id>
```

Format for type 2 is
```
   <type> 1 <HH Id>
```

Format for types 3 and 4 is
```
   <type> 5 <HH Id> <Person Id> <Activity Id> <Start Time> <End
Time>
```

5.3 Library Files

Table 5 lists the activity library files.

Table 5. Activity library files.

Type	File Name	Description
Binary Files	libTIO.a	TRANSIMS Interfaces library
Source Files	actio.h	Defines activity data structures and interface functions
	activityio.c	Activity interface functions source file

5.4 Configuration File Keys

Appendix B lists the TRANSIMS configuration file keys that specify the location of input and output files and required parameters to run the Activity Generator.

Appendix C lists the TRANSIMS configuration file keys that specify the location of input and output files and required parameters to run the Activity Regenerator.

6. Population Converter

The Activity Generator requires that the synthetic population have demographics that exactly match the variables in the regression tree. The population converter program, *PopConverter*, converts a located TRANSIMS synthetic population with demographics derived from the census into a population with demographics that exactly match the variables from the Portland survey. The households in the TRANSIMS population should have the following optional demographics from the 1990 census data: `R18UNDR, RWRKR89,` and `RHHINC`. Each person in the household should have the following demographics from census data: `AGE, RELAT1, SEX,` and `WORK89`.

The household demographics in the regression tree derived from the Portland survey are: household size, income, age less than 5, age 5 to 17, age 26 to 45, household age, number of workers, and household density. The population converter produces the demographics in the regression tree from the census demographics in the TRANSIMS population. The household density variable is determined from land use data at the located household's home activity location. The land use data for activity locations is part of the TRANSIMS network activity location table. The person demographics from the census data are used, but the demographic names are changed to `RELATE, WORK, GENDER,` and `AGE`.

The population converter program must be changed if other demographic variables are used in the regression tree or if different demographics from the census data are in the synthetic population.

Appendix D lists the TRANSIMS configuration file keys that specify the location of input and output files and required parameters to run the Population Converter.

6.1 Usage

The population converter uses three configuration file keys from a TRANSIMS configuration file.

- `ACT_HHDENSITY_HEADER` is used to specify the column header in the activity location table that contains the household density data.
- `POP_LOCATED_FILE` is used to specify the name of the file containing the located TRANSIMS population.
- `ACT_POPULATION_FILE` is the name of the file where the converted population will be written.

The population converter has one command line argument, which is the name of the TRANSIMS configuration file.

```
% $TRANSIMS_HOME/bin/PopConverter   <configuration file name>
```

7. Trip Table Activity Generator

The Trip Table Activity Generator produces activities from the entries in a trip table. The trip table contains information about the number of trips between zones. Also required is a time table that contains the probability of a trip within a time range.

The Trip Table Activity Generator can be used to quickly generate travelers for research purposes, as well as to produce itinerant travelers and freight trips on a transportation network. In addition to the TRANSIMS activity file, the Trip Table Activity Generator produces a population file containing single-person households for each trip and a TRANSIMS vehicle file.

Activities are generated with a car mode (**wcw** mode string). The vehicle type can be specified using the following TRANSIMS configuration file key:

```
VEH_VEHICLE_TYPE
```

The default value of this key is kAuto (1), but it can be set to produce truck vehicle types for freight trips.

The vehicle subtype can be specified. Use the TRANSIMS configuration file key `VEH_VEHICLE_SUBTYPE`.

7.1.1 Determining Origin and Destination Locations

Trip origins and destinations are chosen by a random selection of the possible activity locations in each zone. A column in the TRANSIMS network Activity Location Table identifies the zone associated with each activity location. The column header in the Activity Location Table (defined by the TRANSIMS configuration file key `ACT_TAZ_HEADER`) is used to determine possible origin/destination locations by matching the value in this column with the zone indexes in the trip table.

7.1.2 Determining Activity Times

Two activities are produced for each trip. The number of trips per time of day is determined from a time table that contains time ranges from 0 to 24 hours and probability of a trip within that range.

- The end time of the first activity is based on these probabilities.
- The starting time of the second activity is 3 hours + the end of the first activity.
- The end time of the second activity is 27.0 hours (3 hours past midnight of the second day).

7.2 Files

Trip Table File contains zone pairs and the number of trips between the zones.

Format:
```
<zone 1> <zone 2> <number of trips>
```

Example:
```
1  2  340
1  3  450
```

This example specifies 340 trips between zone 1 and zone 2 and 450 trips between zone 1 and zone 3.

Time Table File contains the lower and upper bounds of the time range in hours and the probability of a trip within the time range. The probabilities in the table should sum to 1.0. The time ranges in the table must cover a 24-hour period (0 – 24 hours).

Format:
```
<time 1>  < time 2>  <probability>
```

Example:
```
 0.0   5.0  0.1
 5 0  19.0  0.8
19 0  24.0  0.1
```

The probability of a trip between midnight (0.0) and 5 a.m. is 0.1, between 5 a.m. and 7 p.m. is 0.8, and between 7 p.m. and midnight is 0.1.

Appendix A: Activity Data Definitions and Format

Field	Description	Allowed Values
Household ID	Each household has a unique household ID. Each Group Quarters is given one household ID. These numbers are assigned in the population file.	integer
PersonID	Each person is given a unique ID in the population file.	integer
ActivityID	Each activity in the household has a unique ID.	integer >0
Activity Type	Definition of activity types may vary. Meaning of the integer value must be specified for each activity set.	integer: 0 through n: Example: 0 = Home 1 = Work 2 = Shop 3 = School 4 = Visit 5 = Other 6 = Serve Passenger
Activity Priority	A 0 (zero) is an activity of lowest priority; a priority of 9 means the activity must be done.	integer: 0 – 9
Starting Time Lower Bound	Earliest time the activity can start.	decimal
Starting Time Upper Bound	Latest time an activity can start.	decimal
Preferred Starting Time a parameter	The time the Route Planner will use as the best guess for the starting time. If this number is –1, the average of the upper and lower bounds is used.	decimal
Preferred Starting Time b parameter	The time the Route Planner will use as the best guess for the starting time. If this number is –1, the average of the upper and lower bounds is used.	decimal
Ending Time Lower Bound	The earliest time the activity can end.	decimal
Ending Time Upper Bound	The latest time the activity can end.	decimal
Preferred Ending Time a parameter	The time the Route Planner will use as the best guess for the activity ending time. If this number is –1, the average of the lower and upper bounds is used.	decimal
Preferred Ending Time b parameter	The time the Route Planner will use as the best guess for the activity ending time. If this number is –1, the average of the lower and upper bounds is used.	decimal
Duration Lower Bound	Shortest length of the activity.	decimal
Duration Upper Bound	Longest length of the activity.	decimal
Duration a parameter	The Route Planner will use this as the best guess of the activity duration. If this number is –1, the average of the upper and lower bound is used.	decimal
Duration b parameter	The Route Planner will use this as the best guess of the activity duration. If this number is –1, the average of the upper and lower bound is used.	decimal
Mode Preference for Arriving at the Activity	This number represents a grammar string that defines the mode preference to the Route Planner (*wcw*, *wt*, …). The correspondence between integer values and possible grammar strings is contained in an external file.	integer

Field	Description	Allowed Values
Vehicle ID	The vehicle ID for all activities with a mode preference of *private auto*, either as driver or as passenger. This field should be set to –1 for all other modes.	integer
Number of Possible Locations for Activity	The number of possible locations in the List of Locations field if value is 1 or greater. The value 0 is not allowed. If this field is –1, the single value in the List of Locations field is an index into a group of activities.	-1, integer ≥ 1
List of Activity Locations	If the Number of Possible Locations field is 1 or greater, this field contains a list of activity location IDs where an activity may take place. If the Number of Possible Locations field is –1, this field contains a number that is an index into a group of activities.	integer [integer] …
Number of Other Participants in the Activity	The number of others in the population who might participate and use the same transportation (e.g., the same car). The number is 0 if the person is to travel alone to the activity.	integer
List of Other Participants	Person IDs of participants using the same transportation. This field should be present only when the value of the Number of Other Participants field is > 0. The first entry in this list is always the person ID of the driver. For passengers, the driver ID is the only entry in this list. For drivers, the person IDs of the passengers are appended to the first entry in the list, the driver ID.	[integer] [integer] …
Activity Group Number	Every activity for an individual will have a number. Sets of activities that must be done together will have the same number.	integer
Notes	Character string used for annotations; free format (may be blank).	255 characters

Appendix B: Activity Generator Configuration File Keys

Configuration File Key	Description
ACT_ACCESS_HEADER	The user data column header in the network activity location file used to specify access to transit.
ACT_ACTIVITY_TYPE	Used to specify the activity types used by the Activity Generator. The base key must be followed with _n where n is an integer to indicate the n^{th} specification of the activity type (non-negative integer).
ACT_BICYCLE_MODE	The number of the bicycle mode (*wyw*) (integer).
ACT_BLOCKGROUP_HEADER	The user data column header in the network activity location file used to specify block group.
ACT_CAR_MODE	The number of the car mode (*wcw*) (integer).
ACT_CARPOOL_ACTIVITY_TYPE	The number of the car pool activity type (non-negative integer).
ACT_DECISION_TREE_FILE	The name of the file containing regression tree for the Activity Generator.
ACT_DEFAULT_CAR_SPEED	The default speed for automobiles in meters/second (floating point number). Default = 37.5
ACT_DEFAULT_INTRAZONE_TRAVEL_TIME	The default travel time within a zone in seconds (integer). Default = 60
ACT_DEFAULT_TRANSIT_MODE	The number of the default transit mode (*wtw*) (integer).
ACT_DEFAULT_TRANSIT_SPEED	The default transit speed in meters/second (floating point number). Default = 30.5
ACT_END_OF_DAY_TIME_RANGE	The time range in hours for lower and upper bounds of start and end times for the end-of-day activity (positive floating number). Default = 0.75
ACT_HOME_ACTIVITY_TYPE	The number of the home activity type (non-negative integer).
ACT_HOME_DURING_DAY_TIME_RANGE	The time range in hours for lower and upper bounds of start and end times for non-work activities originating at home (positive floating point number). Default = 0.75
ACT_HOME_HEADER	The user data column header in the network activity location file used to specify single family home locations.
ACT_INITIAL_HOME_TIME_RANGE	The time range in hours for lower and upper bounds of start and end times for the initial at-home activity (positive floating point number). Default = 0.75

Configuration File Key	Description
`ACT_LOCATION_HEADER`	Used to specify the header for the activity type for activity locations in the network activity location table. The headers must correspond to the activity types defined with the `ACT_ACTIVITY_TYPE_N` keys. The base key must be followed with `_n` where `n` is an integer to indicate the n^{th} specification of the header.
`ACT_MAGIC_MOVE_MODE`	The number of the magic move mode (*wkw*) (integer).
`ACT_MAX_END_TIME`	The maximum end time for an activity in hours past midnight on the starting day (positive floating point number).
`ACT_MODE_WEIGHT_FILE`	The name of the file containing mode coefficients for the activity types. This must contain a coefficient for every mode and activity type.
`ACT_MULTI_FAMILY_HEADER`	The user data column header in the network activity location file used to specify multifamily home locations.
`ACT_OUT_OF_HOME_TIME_RANGE`	The time range in hours for lower and upper bounds of start and end times for non-work activities that do not originate at home (positive floating point number). Default = 0.75
`ACT_PERSON_DEMOG_AGE_HEADER`	The name of the age demographic header for the persons in the population used by the Activity Generator.
`ACT_PERSON_DEMOG_GENDER_HEADER`	The name of the gender demographic header for the persons in the population used by the Activity Generator.
`ACT_PERSON_DEMOG_RELATION_HEADER`	The name of the relationship demographic header for the persons in the population used by the Activity Generator.
`ACT_PERSON_DEMOG_WORKER_HEADER`	The name of the worker demographic header for the persons in the population used by the Activity Generator.
`ACT_POPULATION_FILE`	The name of the file containing a located synthetic population with household and person demographics that exactly match the variables in the Activity Generator regression tree. This file is output from the population converter program.
`ACT_PROBLEM_FILE`	The name of the file where information about problems that occurred during activity generation will be written. Default = *act.problems*
`ACT_RANDOM_SEED`	The random number seed used by activity generators.

Configuration File Key	Description
ACT_REQUIRED_HH_DEMOG	Used to specify the required household demographics in the synthetic population used by the Activity Generator. The base key must be followed with _n where n is an integer to indicate the n^{th} specification of required demographics. The demographics must exactly match and be ordered the same (1 - n) as the demographic variables in the Activity Generator's regression tree.
ACT_SCHOOL_ACTIVITY_TYPE	The number of the school activity type (non-negative integer).
ACT_SHARED_RIDE_DISTANCE_RANGE	The distance range in meters for matching activity locations for shared rides. Default = 2000
ACT_SHARED_RIDE_TIME_RANGE_MAX	The maximum time range in minutes for matching activities for shared rides. Default = 60
ACT_SHARED_RIDE_TIME_RANGE_MIN	The minimum time range in minutes for matching activities for shared rides. Default = 15
ACT_SURVEY_ACTIVITY_FILE	The name of the file containing activity patterns for the survey households.
ACT_SURVEY_HOUSEHOLD_FILE	The name of the file containing the survey household population and demographics.
ACT_SURVEY_WEIGHTS_FILE	The name of the file containing the relative weights of the survey households.
ACT_TAZ_HEADER	The user data column header in the network activity location file used to specify traffic analysis zone.
ACT_TRACT_HEADER	The user data column header in the network activity location file used to specify census tract.
ACT_TRAVEL_TIMES_FILE	The name of the file containing travel time information between zones.
ACT_TRIP_TABLE_OUTPUT	The name of the file that will be output from the Trip Table Activity Generator.
ACT_TRIP_TABLE_VEHICLE_FILE	The name of the vehicle file that will be output from the Trip Table Activity Generator.
ACT_TRIPTABLE_FILE	The name of the file containing the trip table matrix.
ACT_TRIPTABLE_STARTING_HH_ID	The starting household ID for households generated from the Trip Table Activity Generator.
ACT_TRIPTABLE_STARTING_PERSON_ID	The starting person ID for travelers generated from the Trip Table Activity Generator.
ACT_TRIPTABLE_STARTING_VEHICLE_ID	The starting vehicle ID for vehicles generated from the Trip Table Activity Generator.
ACT_TRIPTIME_FILE	The name of the file containing the time of day trip table data.

Configuration File Key	Description
ACT_WALKING_MODE	The number of the walking mode (*w*) (integer).
ACT_WORK_ACTIVITY_TYPE	The number of the work activity type (non-negative integer).
ACT_WORK_HEADER	The user data column header in the network activity location file used to specify work locations.
ACT_WORK_TIME_RANGE	The time range in hours for lower and upper bounds of start and end times for work activities (positive floating point number). Default = 0.25
ACT_ZONE_HEADER	Used to specify the header for the zone attractors, which must match the activity types (ACT_ACTIVITY_TYPE_N). The base key must be followed with _n where n is an integer to indicate the n^{th} specification of the header.
ACT_ZONE_INFO_FILE	The name of the file containing zone attractor data by activity type for the Activity Generator.
ACTIVITY_FILE	
NET_ACTIVITY_LOCATION_TABLE	The network activity location table name.
NET_DIRECTORY	The directory where the network files reside.
NET_LINK_TABLE	The network link table name.
NET_NODE_TABLE	The network node table name.
ROUTER_BIKING_SPEED	The approximate speed in meters/sec for bicycles.
ROUTER_WALKING_SPEED	The approximate speed in meters/sec for walking trips.
VEHICLE_FILE	The name of the TRANSIMS vehicle file for the population.

Appendix C: Activity Regenerator Configuration File Keys

Configuration File Key	Description
ACT_FEEDBACK_FILE	The file containing a list of travelers and associated commands for activity regeneration.
ACT_PARTIAL_OUTPUT	The name of the file that will be output from partial regeneration of activities.

Appendix D: Population Converter Configuration File Keys

Configuration File Key	Description
ACT_HHDENSITY_HEADER	The column header of the household density values in the network activity location tables.
ACT_POPULATION_FILE	The name of the file containing a located synthetic population with household and person demographics that exactly match the variables in the Activity Generator regression tree. This file is output from the Population Converter program.
POP_LOCATED_FILE	The name of the file containing the located population.

Appendix E: Trip Table Activity Generator Configuration File Keys

Configuration File Key	Description
ACT_TAZ_HEADER	The column header in the network activity location file that contains the zone information. Default = TAZ
ACT_TRIP_TABLE_OUTPUT*	The name of the activity file that will be output from the trip table activity generator.
ACT_TRIPTABLE_VEHICLE_FILE*	The name of the vehicle file that will be output from the trip table activity generator.
ACT_TRIPTABLE_FILE*	The name of the file containing the trip table matrix.
ACT_TRIPTABLE_STARTING_HH_ID	The starting household ID for households generated from trip table matrices. Default = 1
ACT_TRIPTABLE_STARTING_PERSON_ID	The starting person ID for travelers generated from trip table matrices. Default = 1
ACT_TRIPTABLE_STARTING_VEHICLE_ID	The starting vehicle ID for vehicles generator from trip table matrices. Default = 1
ACT_TRIPTIME_FILE*	The name of the file containing the time of day trip table data.
MODE_MAP_FILE*	The name of the TRANSIMS mode file containing mapping between mode strings and integer values. The string "*wcw*" must be in this file.
NET_ACTIVITY_LOCATION_TABLE*	The network activity location table. Must contain a column that has the zone number for the activity locations.
NET_DIRECTORY*	The directory where the network tables reside.
NET_LINK_TABLE*	The network link table.
NET_NODE_TABLE*	The network node table.
NET_PARKING_TABLE*	The network parking table.
NET_PROCESS_LINK_TABLE*	The network process link table.
NET_TRANSIT_STOP_TABLE*	The network transit stop table (may be an empty table).
POP_TRIPTABLE_FILE*	The name of the population file that will be output from the trip table activity generator.
VEH_VEHICLE_SUBTYPE	The subtype of the vehicle fleet will be generated. Default = 0
VEH_VEHICLE_TYPE	The type of vehicles that will be generated. Default value is assigned from a type enumeration in the TRANSIMS Network = 1 (kAuto).

* Required configuration file keys. All others are optional and will use default values.

Chapter Three: Index